Corrigé Code de partage avec Capytale : a2c3-8736828

On utilisera les bibliothèques suivantes :

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Exercice 1 - un jeu de données pour s'échauffer

On va étudier les deux séries statistiques u et t :

```
u = [1.06,0.44,2.25,3.88,0.61,1.97,3.43,2.10,1.50,1.68,2.72,1.35,2.94,2.78,3.43,3.58]
t = [2.58,2.25,2.90,3.36,2.41,2.79,3.32,2.81,2.62,2.70,3.17,2.65,3.07,3.13,3.07,3.34]
```

1. Déterminer les moyennes et les écarts-type de ces séries statistiques (on pourra utiliser les fonctions prédéfinies dans `numpy`). On cherchera à créer des variables pour conserver les résultats.

   On peut poser : `mu=np.mean(u), su=np.std(u), mt=np.mean(t), st=np.std(t)`

2. Ecrire un programme qui permet de déterminer la covariance de u et t (on cherchera à conserver le résultat).

   On applique « à la main » la formule $\mathrm{Cov}(u,t) = \sum_{k=1}^{n}(u_k - \overline{u})(t_k - \overline{t})$ (où n est le nombre d'éléments de chacune des listes) en utilisant les résultats précédents :

   ```
   cov=1/len(u)*sum([(u[i]-mu)*(t[i]-mt) for i in range(0,len(u))])
   ```

3. Déterminer le coefficient de corrélation linéaire de u et t

   Par définition, $r_{ut} = \dfrac{\mathrm{Cov}(u,t)}{s_u s_t}$ il suffit donc d'utiliser les résultats précédents : `r=cov/(su*st)`
   et on trouve un résultat supérieur à $0,97$ ce qui signifie que la corrélation est très forte et donc que la modélisation par une relation affine entre u et t est pertinente.

4. Déterminer la droite de régression linéaire reliant u et t

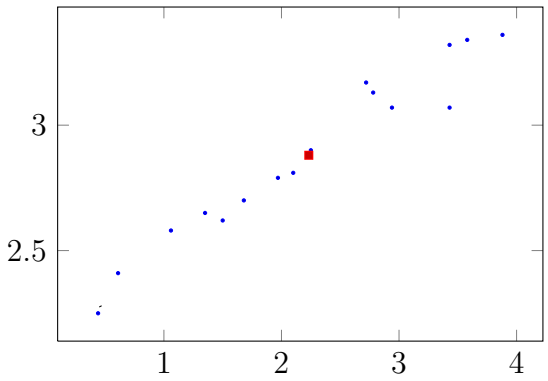
   A nouveau d'après les résultats du cours, on chercher les coefficients a et b de la droite $t = au+b$, où $a = \dfrac{\mathrm{Cov}(u,t)}{s_u^2}$ et $b = \overline{t} - a * \overline{u}$ d'où la commande : `a=cov/su**2 , b=mt-a*mu`
   on obtient $a \simeq 0,31$ et $b \simeq 2,20$

5. Représenter graphiquement le nuage de points de la série statistique (u,v) ainsi que la droite de régression linéaire.

   Il s'agit simplement de représenter le nuage de points, ce que l'on pouvait faire depuis le début, et la droite avec les coefficients trouvés précédemment. Pour la droite, leux points $(0,b)$ et $(4,4a+b)$ suffisent (par défaut Python relie ces deux points « extrêmes »). N.B. : pour le deuxième `plot`, la première liste est celle des abscisses et la deuxième, celle des ordonnées.

   ```
   plt.plot(u,t,'.')
   plt.plot([0,4],[b,4*a+b])
   plt.show()
   ```

   On trouve le graphique suivant (le même que celui illustrant la régression linéaire dans le cours, sans le point moyen, que l'on peut éventuellement ajouter).



Exercice 2 - avec un vrai jeu de données

On va étudier ici une possible corrélation entre le produit intérieur brut et l'espérance de vie à partir de données sur l'année 2020 pour 208 pays (données issues du *World Bank Group*).
Dans un premier temps, il faut importer le jeu de données. On commence donc par exécuter la commande suivante :

```
import pandas as pd
donnees=pd.read_csv('donnees_world_bank.csv', delimiter=';')
```

On rappelle quelques commandes possibles avec pandas : `.head()`, `.describe()`, `.mean()`, `.std()`, `donnees['nom_ colonne']`
On procède ensuite comme à l'**exercice 1**.

Il s'agit exactement des mêmes questions que pour l'**exercice 1** et donc les calculs sont analogues, ce qui change ici est le format des données à la base. Ici, on utilise la bibliothèque `pandas` pour exploiter un fichier (.csv) de données et c'est l'occasion de revoir quelques commandes propres à ce type de données qui se présentent comme un tableau (cf. le `donnees.head()` qui permet d'avoir un aperçu, après avoir importé le fichier).

1. Déterminer les moyennes et les écarts-type de ces séries statistiques (on pourra utiliser les fonctions prédéfinies dans `numpy`). On cherchera à créer des variables pour conserver les résultats.

   La commande `.describe()` fournit toutes les réponses, mais pour stocker les valeurs, on utilisera plutôt les commandes suivantes (`donnees['Life']` donne les valeurs de la colonne `'Life'`, avec en plus une colonne d'indices de ligne). La création de `L` et `G` ci-dessous n'est pas indispensable mais facilite les commandes.

   ```
   L=donnees['Life']
   mL=L.mean()
   sL=L.std()
   G=donnees['GDP_capita']
   mG=G.mean()
   sG=G.std()
   ```

2. Ecrire un programme qui permet de déterminer la covariance des deux variables à étudier (on cherchera à conserver le résultat).

   A nouveau, une des difficultés du format de données est « d'atteindre » les données, par exemple d'une colonne. Pour cela, on peut utiliser la commande suivante :

   ```
   for i in L.index :
       print L[i]
   ```

   Les indices ici vont de 0 à 207 et on aurait pu utiliser un `range(0,208)`, mais ces indices proviennent du fichier d'origine et il peut être dangereux de préjuger de leur organisation.

D'autant plus si on travaille sur un extrait du jeu de données (qui peut ne pas contenir certaines lignes).
Dès lors, il s'agit exactement de la même commande qu'à l'**exercice 1** (en changeant les lettres).

```
covLG=1/len(L)*sum([(L[i]-mL)*(G[i]-mG) for i in range(0,len(L))])
```

3. Déterminer le coefficient de corrélation linéaire des deux variables à étudier.

   ```
   rLG=covLG/(sL*sG)
   ```

   On trouve $0,66$ en valeur approchée, ce qui signifie que la modélisation par une fonction affine, d'une relation entre l'espérance de vie et le produit intérieur brut par personne, n'est pas pertinente.

4. Déterminer la droite de régression linéaire reliant les deux variables à étudier.

   On cherche ici plutôt à voir si l'espérance de vie dépend du produit intérieur brut par personne, on va donc représenter `L` en fonction de `G`. D'où la commande : `aLG=covLG/sG**2;bLG=mL-aLG*mG`

5. Représenter graphiquement le nuage de points de la série statistique des deux variables à étudier ainsi que la droite de régression linéaire.

   Comme plus haut en prenant 0 et 160 000 comme intervalle de représentation :

   ```
   plt.close()
   plt.plot(G,L,'.')
   plt.plot([0,160000],[bLG,aLG*160000+bLG])
   plt.show()
   ```

   Comme nous l'indiquait le coefficient de corrélation linéaire, la modélisation affine n'est pas pertinente. On le voit d'emblée, ne serait-ce qu'en représentant le nuage de points. La droite de régression le confirme à nouveau.
   Par contre en étudiant `L` et le logarithme des valeurs de `G`, on trouve un coefficient de corrélation linéaire de $0,85$ ce qui indique une modélisation affine plus pertinente.